

ABSTRACT

A dynamic race detection system and method overcomes drawbacks of previous lockset approaches, which may produce many false positives, particularly in the context of thread fork/join and asynchronous calls. For each shared memory location, a set of locks that are protecting the location and a set of concurrent thread segments that are accessing the location are maintained. To maintain these sets, each thread maintains a set of locks it is currently holding and a set of thread segments ordered before its current thread segment. Each thread also maintains a virtual clock that is incremented when it forks a second thread. A thread segment is a pair comprising a thread identifier and a virtual clock value. A data race is reported when the lockset for a particular shared memory location is empty and the cardinality of the set of concurrent threads for that memory location is greater than one.